DATA PROCESSING SYSTEM AND METHOD FOR PASSWORD PROTECTING A BOOT DEVICE

CROSS-REFERENCE TO RELATED APPLICATION

Sub a1

~~This application is related to co-pending U.S. Application Serial No. 09/206,686,~~ filed December 7, 1998, which is assigned to the assignee of the present application and ~~incorporated herein by reference.~~

BACKGROUND OF THE INVENTION

1. Field of the Invention:

The present invention relates in general to data processing systems and, in particular, to a data processing system and method for maintaining security while booting the data processing system. Still more particularly, the present invention relates to a data processing system and method for password protecting the boot of the data processing system.

2. Description of the Related Art:

A typical personal computer (PC) system includes a central processing unit (CPU), volatile and non-volatile memory, a display, a keyboard, one or more disk drives, a CD-ROM drive, a pointing device such as a mouse, and an optional network interface card. One of the distinguishing characteristics of PCs is the use of a motherboard or system planar to electrically interconnect these components. Commercially available examples of PCs include the Aptiva™ and Thinkpad™ series of computers available from International Business Machines of Armonk, New York.

The startup software of a conventional PC includes Power-On Self-Test (POST) software to initialize the system components and Basic Input/Output System (BIOS) software to interface the keyboard, mouse and other peripherals. The BIOS software includes a configuration routine that permits a user to select an order in which potential boot devices are checked by the BIOS at startup for an operating system (OS), as well as an OS loader routine that loads the OS from the boot device. For currently available PCs, the list of potential boot devices is generally limited to the hard disk, floppy disk and CD-ROM drives and, optionally, the network interface card.

When unattended, a conventional PC is vulnerable to use by unauthorized user to access confidential information either stored within the PC itself or accessible to the PC through a network. Conventional operating system password protection is relatively ineffective in preventing unauthorized use of a PC because, absent some security mechanism, an unauthorized user can simply use the BIOS configuration routine to select a boot device of choice (e.g., a floppy disk or CD-ROM drive) and boot the PC from the selected boot device utilizing the unauthorized user's own software.

In view of such security concerns, some PCs implement password protection for the BIOS configuration routine so that an unauthorized user cannot change the order in which devices are checked by BIOS for an operating system. Thus, if an operating system is detected on the hard disk drive, an unauthorized user cannot boot the PC from a floppy disk or CD-ROM. The security of a PC may alternatively or additionally be enhanced, as discussed in the above-referenced co-pending application, by requiring a user to enter a password before certain classes of devices can be accessed as a boot device. If necessary, security can be even further enhanced by providing an alarm or lock mechanism to deter removal or opening of the cabinet housing of the PC. Such additional security enhancements deter an unauthorized user from removing the hard disk drive, which may be password protected, and substituting an unprotected hard disk drive in order to gain access to the PC.

The foregoing security precautions have proven effective in preventing an unauthorized user from booting PCs that contain all possible boot devices within their cabinet housing. However, the introduction of new computer interfaces has raised new concerns regarding boot security. For example, the Universal Serial Bus (USB) provides a user accessible interface outside of the cabinet housing of a PC that permits attachment of a large number of peripheral devices. The current commercial USB implementation (i.e., USB 1.1) restricts the devices that may be attached to the USB to fairly low data rate devices, such as printers, cameras, scanners, and floppy disk drives. Because a USB floppy disk drive typically replaces an in-chassis floppy disk drive in the BIOS-defined boot order, existing security mechanisms, such as password protection of the BIOS configuration routine, prevent an unauthorized user from accessing a PC by attaching the user's own USB floppy disk drive and booting from a floppy disk.

However, the present invention recognizes that emerging peripheral connection technologies such as USB 2.0 support much higher data rates and therefore again make a PC vulnerable to unauthorized booting from a USB 2.0 hard disk drive or CD-ROM drive. For example, if a user has a PC that is configured to boot from a USB 2.0-compliant hard disk drive (which may even be password protected), it is a trivial exercise for an unauthorized user to connect his own hard disk drive in lieu of the password protected hard disk drive and access the PC. Moreover, such unauthorized access would be difficult to detect because none of the conventional tamper detection mechanisms would be triggered by swapping USB devices.

Therefore a need exists for a data processing system and method for providing security that prevent an unauthorized boot of a computer.

SUMMARY OF THE INVENTION

A data processing system and method of password protecting the boot of a data processing system are disclosed. According to the method, in response to an attempt to boot the data processing system utilizing a boot device, the boot device is interrogated for a password. If the boot device supplies password information corresponding to that of a trusted boot device, the data processing system boots utilizing the boot device. If, however, the boot device does not supply password information corresponding to that of a trusted boot device, booting from the boot device is inhibited. In a preferred embodiment, the password information comprises a unique combination of the boot device's manufacturer-supplied model and serial numbers.

The above as well as additional objectives, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features are set forth in the appended claims. The present invention itself, however, as well as a preferred mode of use, further objectives, and advantages thereof, will best be understood by reference to the following detailed description of a preferred embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 is a high level block diagram of a data processing system having a boot security mechanism in accordance with the present invention;

Figure 2A is a high level logical flowchart illustrating a password-protected boot process in accordance with a preferred embodiment of the present invention; and

Figure 2B is a high level logical flowchart depicting a BIOS configuration routine utilized to add trusted boot devices to a computer system in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

With reference now to the figures and in particular with reference to **Figure 1**, there is illustrated a high level block diagram of a data processing system in accordance with a preferred embodiment of the method and system of the present invention. Although those skilled in the art will recognize that the depicted data processing system is a personal computer system (PC), it should be appreciated that the present invention is not limited to PCs, but is also applicable to other data processing systems.

As shown in **Figure 1**, computer system **10** has a system bus **11** connected to a central processing unit (CPU) **12**, which executes software instructions and controls the operation of computer **10**. During operation, CPU **12** utilizes system bus **11** to access data and instructions within read-only memory (ROM) **13**, which stores startup software such as POST and BIOS, and dynamic random access memory (DRAM) **14**, which provides storage for operating system and application data and instructions. System bus **40** is coupled to a Peripheral Component Interconnect (PCI) local bus **20** via PCI host bridge **16**. PCI host bridge **16** provides both a low latency path through which CPU **12** may directly access PCI devices mapped to bus memory and/or I/O address spaces and a high bandwidth path through which PCI devices may directly access DRAM **14**.

The PCI devices connected to PCI local bus **20** include a disk adapter **18**, which provides an interface for a hard disk drive **19**, and a network interface card (NIC), which provides a wired or wireless interface to a communication network **17**. In order to present audio and video data to a user, computer system **10** is further equipped with a PCI-compatible audio controller **23** and graphics controller **21**, which drive stereo speakers **24** and display device **22**, respectively.

PCI bus **50** is further coupled to an expansion bus, such as ISA bus **25,** via expansion bus bridge **29**. Coupled to ISA bus **25** via an unillustrated I/O controller are conventional

input devices, such as a keyboard **26** and mouse **28**. Other peripheral devices, such as CD-Rewritable (CD-RW) drive **30** (as well as cameras, printers, hard and floppy disk drives, etc.) can be interfaced to PCI local bus **20** via USB bridge **34** and an externally accessible port of Universal Serial Bus (USB) **32**. In a preferred embodiment, USB **32** supports USB 2.0, which allows a data transfer rate of 480 Mbit/s, thus making it convenient to connect boot devices external to the cabinet housing of computer system **10**. More information regarding USB 2.0 can be found in the Universal Serial Bus Revision 2.0 specification, which is available from the USB Implementers Forum, Inc., and is incorporated herein by reference.

Referring now to **Figure 2A**, there is illustrated a high level logical flow chart of the startup of a computer system having a password protected boot sequence in accordance with the method and system of the present invention. The process depicted in **Figure 2A** begins at block **100**, for example, in response to power on or power-on-reset (POR) of computer system **10**. The process then proceeds to block **102**, which illustrates CPU **12** executing POST software out of ROM **13** so that the components of computer system **10** are placed in a known, stable state. Next, as shown at block **104**, CPU **12** begins execution of BIOS software, for example, to interface key peripherals, such as keyboard **26**, mouse **28**, and display **22**. As depicted at block **106**, the BIOS software determines whether a request to enter the BIOS configuration routine has been received. A user may request to enter the BIOS configuration routine, for example, by depressing a designated function key (e.g., F1) of keyboard **26** during the execution of the BIOS software. If no request to enter the BIOS configuration routine is received, the process passes to block **120**, which is described below. However, if a BIOS configuration request is received, the BIOS software prompts the user to enter a configuration password. As described above, entry to the BIOS configuration routine is preferably password protected to prevent unauthorized changes to the order (priority) in which boot devices are checked for a bootable operating system at system startup. If the user does not enter the correct configuration password, the process passes to block **120**, which is described below. If, however, the user enters the correct configuration

password, the process proceeds from block **108** through page connector A to the BIOS configuration routine illustrated in **Figure 2B**.

With reference now to **Figure 2B**, there is illustrated a high level logical flowchart of a BIOS configuration routine utilized to add and prioritize trusted boot devices of a computer system in accordance with a preferred embodiment of the present invention. Following page connector A, the process passes to block **110**, which illustrates selection of a boot device to add as a trusted boot device from which computer system **10** will be allowed to boot. The user may select the boot device, for example, by utilizing mouse **28** or keyboard **26** to select from among a menu of boot devices displayed in a dialog box within display **22**. Next, as illustrated, at block **112**, the BIOS configuration routine interrogates the selected boot device for a unique device password that will be utilized during startup to verify that the boot device is a trusted device from which computer system **10** is permitted to boot.

In a preferred embodiment, the unique device password for the boot device is a combination of the model and serial number of the boot device. As will be appreciated by those skilled in the art, most manufacturers of devices that can serve as boot devices (e.g., optical, hard disk, floppy disk, and Zip™ drives) store the manufacturer's name, device model number and device serial number in either one-time programmable read-only memory (OTPROM) or electrically erasable programmable read-only memory (EEPROM) in the device. Many bootable devices are designed to supply such information in response to commands, such as the USB 2.0 "Get Device Descriptors" command or similar commands within the Integrated Device Electronics (IDE), Serial IDE and SCSI command sets. Because modification of the model and serial numbers require specialized knowledge and equipment (and possibly disassembly of the bootable device) and is therefore beyond the capabilities of most individuals, the use of the manufacturer-specified model and serial numbers as a password offers a reasonable level of security.

After the boot device selected for addition to the boot sequence has provided the BIOS configuration routine with a unique password, the BIOS configuration routine stores the unique password in non-volatile storage at block **114**, preferably after hashing the password with a selected encryption algorithm. The hashed password may be stored, for example, in non-volatile RAM (NVRAM), in a security chip, or on hard disk drive **19**. As illustrated at block **115**, the newly added boot device is then assigned a priority in the boot sequence either by the user or by the BIOS configuration routine. A determination is then made at block **116** whether or not the user wishes to set up an additional boot device, for example, by prompting the user with a dialog box displayed within display **22**. If so, the process returns to block **110-115**, which have been described. If, however, the user does not wish to set up an additional boot device, the BIOS configuration routine exits and returns to block **120** of **Figure 2A** through page connector B.

Referring again to **Figure 2A**, block **120** illustrates the BIOS software determining the priority of boot devices present in computer system **10** and the password requirement of each boot device, if any. As shown at blocks **122-134**, the BIOS software then scans through the list of boot devices in sequence from the highest priority device to the lowest priority device to locate the highest priority device from which computer system **10** can boot an operating system. Thus, at block **122**, the BIOS software selects the highest priority boot device that has not been checked for an operating system from the list of possible boot devices. Next, block **124** illustrates the BIOS software determining whether or not the selected boot device is capable of booting an operating system. If not, the process returns to block **122**, where the boot device having the next highest priority is selected. If, however, a determination is made at block **124** that the selected boot device is capable of booting the computer system, the process passes to block **126**, which depicts the BIOS software determining whether or not a correct entry of a password is required to boot from the selected boot device. If a password in not required (e.g., because the selected boot device is in-cabinet hard disk drive **19**), the process passes to block **132**, which illustrates booting an

operating system for computer system **10** from the selected boot device. Processing thereafter continues at block **134** under the control of the operating system.

Returning to block **126**, in response to a determination that entry of a password is required to boot from the selected boot device, the process proceeds to block **128**, which depicts the BIOS software interrogating non-volatile storage in the selected boot device for a device password. For example, if the selected boot device under consideration is CD-RW drive **30**, the step illustrated at block **128** may entail sending CD-RW device **30** a USB 2.0 "Get Device Descriptors" command, as discussed above. The BIOS software then determines at block **130** whether or not the correct password was entered, for example, by hashing a string formed by concatenating the boot device's model and serial numbers and comparing the resulting hash with the stored hashes of one or more trusted boot devices. If a correct device password was not entered, the process returns to block **122**, thereby signifying that the device from which a boot was attempted is not a trusted device from which computer system **10** is permitted to boot. If, however, a determination is made at block **130** that the correct password was obtained from the boot device, the process passes to block **132**, which illustrates the BIOS software booting an operating system utilizing the selected boot device. Thereafter, processing continues under the control of the operating system at block **134**.

As has been described, the present invention provides an improved method and system for password protecting the boot of a computer system. In accordance with the present invention, before a boot device is permitted to boot the data processing system, the data processing system interrogates the boot device for a password corresponding to a trusted boot device. If the boot device supplies a password corresponding to a trusted device, then the boot device is permitted to boot the data processing system. If the boot device fails to supply a password corresponding to that of a trusted device, then the boot device is not permitted to boot the data processing system. In this manner, boot security of the data is enhanced with password protection that cannot easily be circumvented by an unauthorized

users who connects his own boot device to a USB port or other externally accessible connector of the data processing system.

While a preferred embodiment has been particularly shown and described, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the present invention. For example, although the present invention has principally been described with reference to an embodiment that protects a computer system from booting from an unauthorized USB boot device, the present invention is not limited to such embodiments, but is instead applicable to other externally connectable boot devices including those complying with the IEEE 1394 (also referred to by the trademarks FireWire™ or i.Link™) standard. Moreover, although aspects of the present invention have been described with respect to a computer system executing software that directs the functions of the present invention, it should be understood that present invention may alternatively be implemented as a program product for use with a data processing system. Programs defining the functions of the present invention can be delivered to a data processing system via a variety of signal-bearing media, which include, without limitation, non-rewritable storage media (e.g., CD-ROM), rewritable storage media (e.g., a floppy diskette or hard disk drive), and communication media, such as digital and analog networks. It should be understood, therefore, that such signal-bearing media, when carrying or encoding computer readable instructions that direct the functions of the present invention, represent alternative embodiments of the present invention.